

IN THE CLAIMS:

1. (Currently amended) A method for identifying non-externalized strings that are not hard-coded comprising the computer-implemented steps of:
 scanning a code for a first pair of string delimiters; [[and]]
 determining whether a string within said first pair of string delimiters is a path name to a resource file; and
 ~~wherein if said string is not a path name to said resource file then~~ flagging said string is a non-externalized string that is not hard-coded as a possible hard-coded string.
2. (Currently amended) The method as recited in claim 1 ~~further comprising the step of:~~
 ~~identifying~~ wherein said string is not flagged as a possible hard-coded string if said string is [[not]] a path name to said resource file.
3. (Currently amended) The method as recited in claim 1, wherein said code comprises [[Java]] platform-independent byte code.
4. (Original) The method as recited in claim 1, wherein said path name is a uniform resource locator.
5. (Original) The method as recited in claim 1, wherein said path name is a resource locator.
6. (Original) The method as recited in claim 1, wherein said string within said first pair of string delimiters is a path name to said resource file if said string is in a dot delimited notation.
7. (Original) The method as recited in claim 1, wherein said code is scanned line by line until said first pair of string delimiters is identified.
8. (Original) The method as recited in claim 7, wherein if there is any more code to be scanned after said first pair of string delimiters is identified, then the method further comprises the step of:
 continuing to scan said code for a second pair of delimiters.

9. (Currently amended) A computer program product in a computer readable medium for identifying non-externalized string that are not hard-coded, comprising:
programming operable for scanning a code for a first pair of string delimiters;
[[and]]
programming operable for determining whether a string within said first pair of string delimiters is a path name to a resource file; and
programming operable for flagging said string as possible hard-coded string
~~wherein if said string is not a path name to said resource file then said string is a non-externalized string that is not hard-coded.~~
10. (Currently amended) The computer program product as recited in claim 9 ~~further comprises:~~
~~programming operable for identifying wherein said string is not flagged as a possible~~
hard-coded string if said string is [[not]] a path name to said resource file.
11. (Currently amended) The computer program product as recited in claim 9, wherein said code comprises [[Java]] platform-independent byte code.
12. (Original) The computer program product as recited in claim 9, wherein said path name is a uniform resource locator.
13. (Original) The computer program product as recited in claim 9, wherein said resource file is a resource bundle.
14. (Original) The computer program product as recited in claim 9, wherein said string within said first pair of string delimiters is a path name to said resource file if said string is in a dot delimited notation.
15. (Original) The computer program product as recited in claim 9, wherein said code is scanned line by line until said first pair of string delimiters is identified.

16. (Original) The computer program product as recited in claim 15, wherein if there is any more code to be scanned after said first pair of string delimiters is identified, then the method further comprises:

programming operable for continuing to scan said code for a second pair of delimiters.

17. (Currently amended) A data processing system, comprising:

a processor; and

a memory unit for storing instructions of said processor;

an input mechanism;

an output mechanism;

a bus system for coupling the processor to the memory unit, input mechanism, and output mechanism;

means for scanning a code for a first pair of string delimiters; [[and]]

means for determining whether a string within said first pair of string delimiters is a path name to a resource file; and

wherein means for flagging said string as a possible hard-coded string if said string is not a path name to said resource file then said string is a non-externalized string that is not hard-coded.

18. (Currently amended) The data processing system as recited in claim 17, wherein the system further comprises:

means for ~~identifying~~ not flagging said string as a possible hard-coded string if said string is [[not]] a path name to said resource file.

19. (Currently amended) The data processing system as recited in claim 17, wherein said code comprises [[Java]] platform-independent byte code.

20. (Original) The data processing system as recited in claim 17, wherein said path name is a uniform resource locator.

21. (Original) The data processing system as recited in claim 17, wherein said resource file is a resource bundle.

22. (Original) The data processing system as recited in claim 17, wherein said string within said first pair of string delimiters is a path name to said resource file if said string is in a dot delimited notation.

23. (Original) The data processing system as recited in claim 17, wherein said code is scanned line by line until said first pair of string delimiters is identified.

24. (Original) The data processing system as recited in claim 23, wherein if there is any more code to be scanned after said first pair of string delimiters is identified, then the method further comprises:

means for continuing to scan said code for a second pair of delimiters.